

# Swarmrobot.org – Open-hardware Microrobotic Project for Large-scale Artificial Swarms

Serge Kernbach

Institute of Parallel and Distributed High-Performance Systems,  
University of Stuttgart, Universitaetsstr. 38, 70569 Stuttgart, Germany,  
Serge.Kernbach@ipvs.uni-stuttgart.de

**Abstract.** The purpose of this paper is to give an overview of the open-hardware microrobotic project swarmrobot.org and the platform Jasmine for building large-scale artificial swarms. The project targets an open development of cost-effective hardware and software for a quick implementation of swarm behavior with real robots. Detailed instructions for making the robot, open-source simulator, software libraries and multiple publications about performed experiments are ready for download and intend to facilitate exploration of collective and emergent phenomena, guided self-organization and swarm robotics in experimental way.

## 1 Introduction

The high miniaturization degree of the robotic systems with increasing efficiency of the hardware and software represents an important trend in the area of collective systems [1]. The subarea – swarm robotics – is an emerged research field focused on designing collective intelligent systems comprised by a large number of robots. Its fascination and theoretical foundations originate from studying and understanding a group behavior of animals and insects societies [2]. It is expected that in a similar way a group of relatively simple and cheap robots can solve complex tasks that are beyond the capabilities of a single robot [3].

Natural physical and biological systems, being a metaphor for the phenomenon of self-organization, can organize themselves to emerge complex macroscopic behavior or spatiotemporal ordered structures and formations [4]. There is no central element coordinating the system, a group behavior (macroscopic level) emerges from individual set of local rules (microscopic level) via interactions and cooperation. However, natural systems have multiple capabilities of interactions, perception and communication, which are significantly more complex than those in technical systems. Microrobots, due to a small size, are very restricted in locomotion, sensing and communication. Therefore a swarm-like behavior, expected from microrobotic systems, can "approach" a complexity of natural phenomena only when the robots are specifically developed, equipped and highly optimized for the desired collective activities [5].

Being motivated by the challenge of creating artificial self-organizing and emergent phenomena, we started open-hardware microrobotic research project

for developing a specialized robot platform for large-scale swarms. It is based on the concept of swarm embodiment and is completely open at [www.swarmrobot.org](http://www.swarmrobot.org), see Fig. 1, for contributions from the domains of electrical and mechanical engineering, computer science and biology. The project consists of several parts:



**Fig. 1.** *Swarmrobot.org*, which is devoted to the platform Jasmine and large-scale swarm experiments.

a development of cheap and reliable hardware platform, an integrated SDK, open-source simulator and a description of the preformed experiments. All of them should allow a quick implementation of swarm behavior with real robots, exploration of collective phenomena emerged by artificial self-organization and may be of interest for broad areas of research and education. Since the project is maintained by universities, we can neither manufacture the robots for other participants nor perform any commercial activities with the platform – the project [swarmrobot.org](http://swarmrobot.org) targets primarily scientific goals.

This paper is structured in the following way: Section 2 briefly describes the concept of swarm embodiment, which underlies the development. Sections 3, 4 and 5 survey hardware, software and simulator for the robot platform. Finally, in Section 6 we overview performed experiments with Jasmine robots.

## 2 Concept of swarm embodiment

Swarm embodiment represents one of the main motivations for starting an open-hardware microrobotic development. The origin of this concept lies in the phenomenon of artificial self-organization (SO). Artificial SO differs from natural one in a couple of essential points, where a purposeful character of SO (so-called guided self-organization) represents the most important aspect. It means that a developer creates artificial SO to achieve some desired emergent effects, such as a specific group behavior of robots or desired molecular self-assembled structures.

Artificial SO is created by local rules. Each participant, driven by these local rules, demonstrates an individual behavior. All individual behaviors are macroscopically observable in the form of a coherent collective behavior. From the viewpoint of an observer, the collective behavior is emerged from interactions among swarm agents, which are guided by local rules. Derivation of local rules represent a serious problem, which is analytically non-solvable as proved by Poincare for the physical case of N-bodies interaction. There are three main strategies to derive (to approximate) such rules in non-analytical way. At the *bottom-up strategy*, the local rules are first programmed into each agent [6]. Performing many simulations and gradually changing the local rules, a desired collective behavior can be derived. Using the *top-down strategy*, the derivation of local rules starts from a definition of the macroscopic pattern and the corresponding constraints. Then, using "distributing" transformation or evolutionary strategies, this macroscopic pattern can be transformed into a set of local rules, that in turn generate the desired pattern [7]. The last way consists in *observing the behavior* of already existed artificial and natural swarm-like systems and in trying to reproduce this behavior [2]. All these ways lead finally to the local rules that control the behavior of each swarm agent.

The researcher, once derived these local rules and intended to implement them in a real robotic or in a simulation system, can face the questions "*which general degree of collective intelligence is feasible in the destined system?*" and "*how to implement it with the obtained local rules?*" To illustrate this point, we collect in Table 1 several swarm activities that robots can collectively perform. We can roughly say, that these collective activities represent some building blocks for purposeful design of the swarm intelligence. Let us take the most simple example of spatial orientation and assume a robot has found a "food source" being relevant for the whole swarm. This robot, guided by local rules, sends the following message: "I, robot X, found Y, come to me". Other robots, receiving this message can propagate it further through a swarm, so that finally each robot knows "there is a resource Y at the robot X". However, none of robots can find it because they do not know a coordinate of this "food source". The robot X cannot provide these coordinates because it does not know its own position. In this way, even possessing corresponding local rules and capabilities to communicate, robot cannot execute the desired collective activity "find food source" without some additional efforts intended to localization. Considering the localization problem, we face in turn the next generation of questions "*how to implement it?*" In this

**Table 1.** Several types of collective activities performed by a swarm.

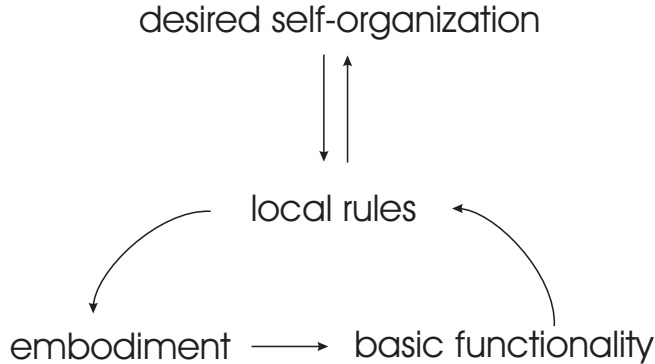
Context	N	Swarm Capability
Spatial	1	Spatial orientation
	2	Building spatial structures
	3	Collective movement
Information	4	Building informational structures
	5	Collective decision making
	6	Collective information processing
	7	Collective perception/recognition
Functional	8	Building functional structures
	9	Collective task decomposition
	10	Collective planning
	11	Group-based specialization

way, the original problem of local rules, generating a desired self-organization, changes into the problem of their implementation in real systems.

More generally, during a derivation of local rules  $R_k$  we assume some basic functionality  $F_b$ , like message transmission, localization or environmental sensing. Often we do not take into account real restrictions underlying this basic functionality or they are generally unknown at this step. However, implementing later these  $R_k$ , we obtain the swarm behavior, that differs from our expectations (even with correctly derived  $R_k$ ), because a real functionality  $F_b$  can essentially differ from the expected one.

To get round this problem, we involved the embodiment concept (see Fig. 2). This says the same functionality can be implemented in many different ways: "intelligent behavior" can even be implemented when using only some properties of materials [8]. Embodied functionality is also often implemented in some "unusual" way. For example a robot can estimate a distance to neighbors by sending an IR-impulse and measuring a reflected light. However, distances can also be obtaining during communication by measuring a signal intensity. This simple mechanism saves time and energy: such an unusual functionality is a typical sight of embodiment.

The embodiment in our context means the system possesses the desired functionality  $F_b$ , but this functionality is in a latent form, "it is not appeared". This offers a way of how to get basic functionality for the local rules  $R_k$ : the local rules have to influence the hardware development of a robot. The swarm embodiment takes then the following form: definition of the macroscopic pattern  $\Omega$  and the corresponding microscopic/macroscopic constraints; derivation of the local rules  $R_k$ ; trade off between required functionality and adjustment of hardware; change of the hardware. The local rules have always been considered as a pure software components, however now they are a combination between software and hardware. We can say that in this way *the local rules for the whole swarm behavior are embodied into each individual robot.*



**Fig. 2.** Concept of embodiment of local rules.

The embodiment in sense "*hardware*  $\rightarrow$  *rules*" has been demonstrated e.g. in works [7] [9], where we analyzed a dependence between agent's movement and sensor data for the derived  $S_k$ . Optimizing local rules to specific motion system, the "top-down"-derived rules can be 5-20% more efficient than corresponding "bottom-up" rules. The embodiment in sense "*rules*  $\rightarrow$  *hardware*" has been demonstrated e.g. in works [10], [11]. In these works we considered context awareness related collective capabilities of interacting robotic group and incorporate several local rules into specific sensor system of real microrobots. The achieved results improve collective robotic behavior and reduce required communication and computational efforts.

After those experiments we came to conclusion that *swarm robot differs from a general-purpose robot*, even when this robot is small enough. The swarm-capable robot should not only incorporate in its own construction some specific details, such as capabilities of swarm communication or cooperative actuation, but be open for embodiment of different local rules. It should have some open-hardware structure. This was the main motivation to derive new microrobotic platform for large-scale swarm experiment and make it in open-hardware manner.

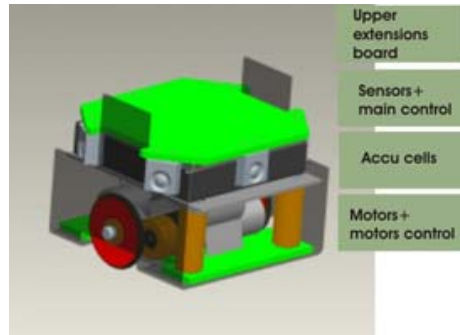
### 3 Open-hardware development

The design of the robot, assembling instructions, electrical schemes and PCB can be found at [www.swarmrobot.org](http://www.swarmrobot.org). Here we briefly sketch several main points:

1. One of the most important requirements is that the microrobot platform for large-scale swarm experiments should be **cheap**. We require that the swarm robot should cost at most 100 euro. Obviously, this requirement imposes several compromises on the capabilities of the platform.

2. The robot should be small, no larger than a cube with the edge of 30 mm. This size hardly limits hardware, so that we come closely to real "limited swarm agent".
3. It should be easy in assembling so that each research institute or even private persons can produce enough robots for their own swarms. This means primarily avoiding SMD soldering and a high-precision micro-mechanical assembling.
4. The robot should have enough hardware resources (ROM, RAM, MPU frequency) to maintain at least a simple operational system with users-defined software.
5. Rich communication and perception capabilities oriented for large-scale swarms. This means: omni-directional communication system; communication radius, which is large enough for a reasonable information transfer in a swarm; support for context-based features; perception system, which allow recognition of main object features such as geometry or colors.
6. The running time without a new recharging should be at least 1-2 hours for the hardest conditions (full speed motion, continuous communication and perception);
7. The robot has to be extendable (modular) for other sensors/actuators/ communication boards with different embodiment rules. This means: standardized inter-board logical, mechanical and electrical interfaces; standardized robots size and structural construction.

General design of the microrobot is limited by the size and geometrical configurations of used components. Since we would like to have a robot that is small as possible, but still cheap and easy in assembling, the most efficient structure in this case is either an integration of all components on one PCB (flexible PCB) or a "sandwich design" (see Fig. 3). Analyzing sensors for sensing and com-



**Fig. 3.** "Sandwich design" of the Jasmine platform.

munication we come at a conclusion that SMD sensors are not really suitable. Thus, an integration of all components into one PCB is not possible and so only

”sandwich design” can be used. Order of ”sandwich layers” is defined by the size of accumulator: it can be placed only between the motor PCB and sensors board. All layers of this ”sandwich” can be covered by a thin plastic or, more sophisticated, by metal chassis. This design is simple for assembling, cheap and protect electronics and mechanics from possible damages.

The robot uses two GM15(RM-N1) motors with gear motor-wheel coupling, one Li-Po accumulator cell  $C=250\text{mA/h}$  with 3.7V, electronics of the robot works with 3V. Robot consumes about 200-220 mA in a full motion and communication modus, that is less than 1C and it has enough capacity for running time of 1-2 hours. The sensing and communication capabilities of the Jasmine platform are already described (see e.g. for communication [10] and for perception [12]). Briefly, the robot can recognize the surface geometry (concave, convex, round, flat), color of surfaces (at least three main colors), can sense obstacles in 360 degree around itself, in a distance over 150mm, internal energy level and so on. Communication is omni-directional, half-duplex, max. communication radius is over 150mm (300mm in 15 degree opening angle). In Fig. 4 we show different developed versions of the platform.

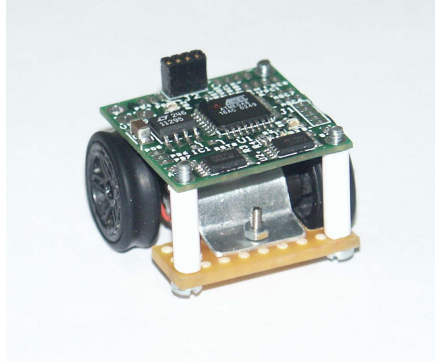
The microrobotic platform is modular, it means the robot can be configured as main board plus different extensions boards (see Fig. 5). Currently there are a few extension systems, listed in the Table 2 that are in different state of development/testing.

**Table 2.** Potential and already existed extension boards for Jasmine platform.

Extension	Description
Ego-position system	Allows determining the robot position/rotation in global coordinate system with accuracy of a few mm. It requires a beamer on the top of robot-arena, software for beamer control and specific two-sensor system on the top of the robot.
Light sensors system	Allows following the light source. It requires two light sensor on the top of the robot.
Wireless communication	Uses Atmel’s RF chip on the top extension. It requires a relatively large antenna.
Odometrical system	Allows determining the local robot displacement and rotation in with accuracy of a few mm. It required specific sensor system on the motor board.
Auto Recharge System	Allows recharging of robot’ accu during a motion. The robot can decide when its own accu needs to be recharged and does it autonomously without human participation.
Electro-magnetic gripper	Allows gripping small metallic objects (in development).

Finally, we summarize the main capabilities of the platform:

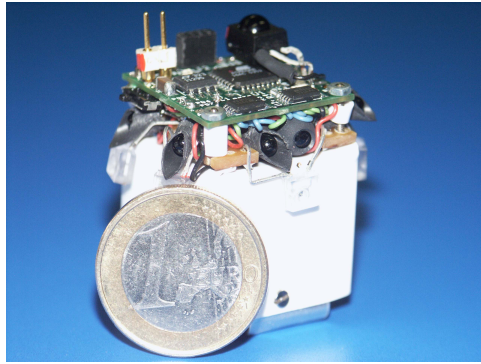
- low-price solution, as components less than 100 euro per robot;



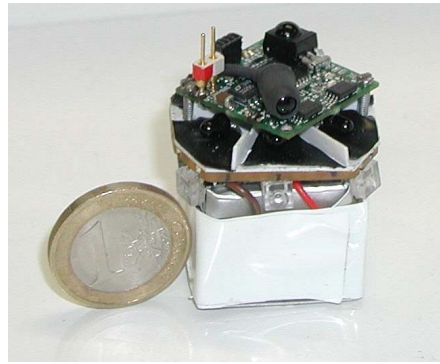
(a)



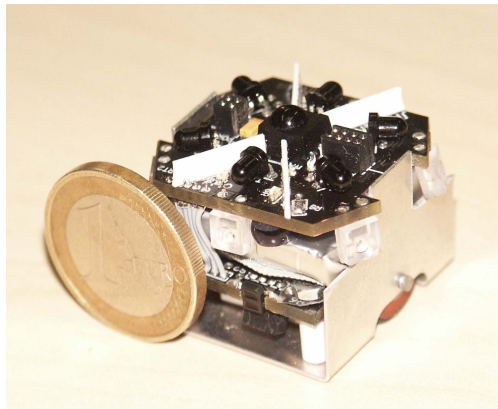
(b)



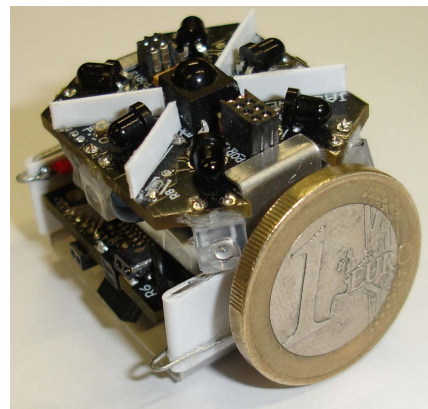
(c)



(d)



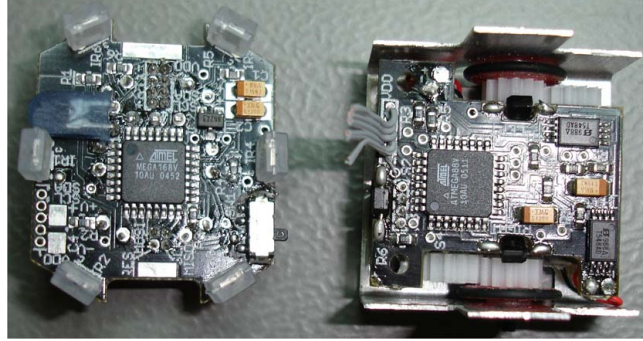
(e)



(f)

**Fig. 4.** Different versions of the microrobot Jasmine. (a) Chassis with the megabitty board; (b) The first test development for testing of sensors; (c) The first version; (d) The second version; (e) The third version; (f) The third plus version;





**Fig. 5.** Modularity of the Jasmine platform: the main board (left) and the motors board (right).

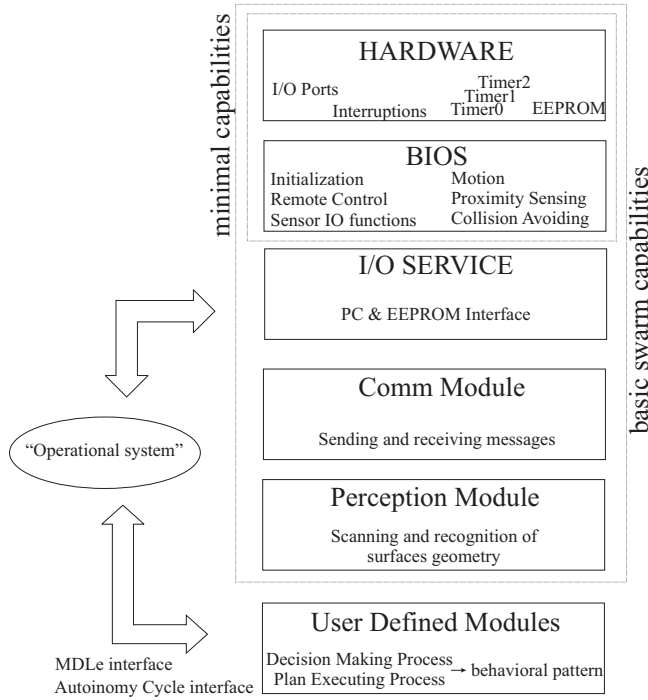
- size 26 x 26 x 26 mm;
- RAM 2kB, Flash ROM 24kB, nonvolatile EEPROM 1kB;
- 2 DC motors (reversible), max. velocity over 500 mm/sec.;
- autonomous work 1-2 hours;
- omni-directional 6 channel robot-robot communication, half-duplex;
- physically adjustable communication radius: 0-max, max. communication over 300 mm, capacity of channel over 1000 bits/sec.;
- remote control and host-robot communication;
- proximity and distance measurement over 300 mm;
- perception of objects geometries, light, touch, internal energy sensor and so on;
- extendable periphery (e.g. camera) with serial interface, external actuators;
- programmable via COM-Port from PC, many open-source development tools.

Since this is the open-hardware project, the hardware details can be downloaded from the site, otherwise in the community there are a few companies that can provide a complete solution with a hardware support.

## 4 Control concepts and supported software

The software is structured hierarchically, where the high-level modules are based on the low-level modules, as shown in Fig. 6. These high-level modules can be exchanged without changing basic capability of the robot. Since the high-level modules describe the behavioral scenarios, we can implement many different scenarios without overstepping hardware capabilities (flash memory) of the micro-robot. The modular construction allows also a "distributed" software development by different persons, easy improvement and extension. Moreover, modules that are not required in a particular scenario, can be removed from a configuration without destroying the software architecture.

There are five main levels of the software architecture:



**Fig. 6.** Software structure of Jasmine platform.

1. BIOS Module: This module provides the main interface to hardware resources. It supports remote control, motion control, proximity sensing, TWI interface, software interruption system, collision avoiding and so represents the basic capabilities of the microrobot.

2. I/O Services: This module provide the interface to PC (serial interface) for reading/writting and writing into EEPROM.

3. Comm Module: Communication module provides a low-level half-duplex local data exchange between the robots. It supports two main protocol: with confirmation and without confirmation. The module is relatively large and can be removed from configuration (in scenarios where communication is not required). This module includes also a support for active communication (requires user-defined support).

4. Perception Module: Perception module is destined for scanning and recognition of surfaces geometry. This routine includes also basic functions for collective perception (requires user-defined modules).

5. User-defined module: This module includes primarily two functions *DecisionProcess()* and *ExecutePlan()* and determines the high-level behavioral patterns. Each user (working on separate scenario) creates its own user-defined module or uses them from share-library.

The "operating system" of the robot executes user-defined code and provides hardware resources, timing and other resources to a user-defined code. There are two operating systems that use either the autonomy cycle or software interruptions. Independently of the used operating system, the BIOS, perception and communication modules remain the same. These components build an integrated on-board SDK, which allows a quick and efficient implementation of swarm algorithms. The C-source and libraries for AVR compiler (open-source) can be downloaded from the site.

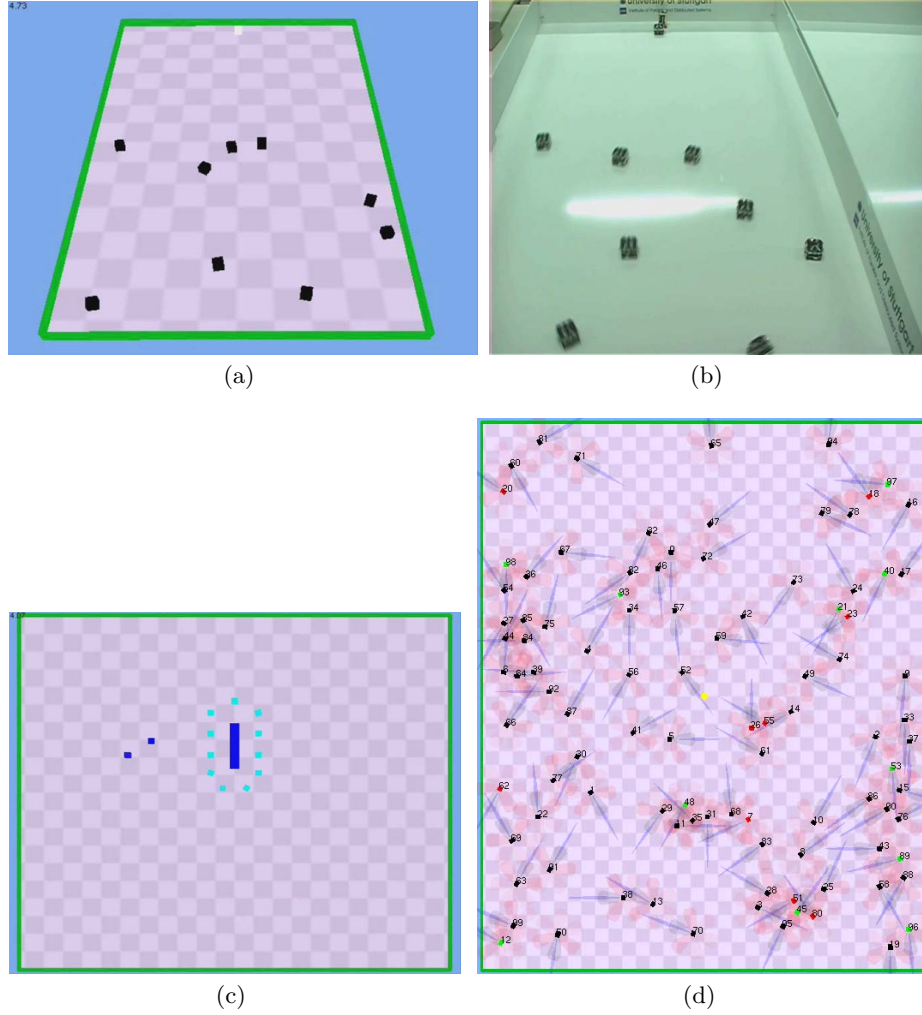
## 5 Simulator

The simulator is used for preliminary tests and debugging the software before programming real robots. The program used to create the simulation system is called 'Breve'. It is an open-source, multiplatform software package, which makes an fast development of 3D simulations for decentralized systems and artificial life. Users define the behaviours of agents in 3D world and observe how they interact. 'Breve' includes an OpenGL engine so it is possible to visualize simulated worlds. Breve simulations are written in an object-oriented language called Steve. More information about Breve can be found in [13]. Breve executes the code written in each simulation agent in a parallel way every iteration, so all agents are independent.

Especial attention was paid to create equal hardware abstraction layers in Jasmine robots and in the simulator so that the same code can be run in both systems without essential modifications. A large effort was invested into making reality gap as minimal as possible: for instance, we compared behavior between simulated and real setups, see Fig. 7(a) and 7(b), and tuned the simulator to obtain almost identical behaviors, see more in [14]. Since all sensors are fully implemented, the simulator was employed even for preliminary tests with collective perception based on surface colors and surface geometries, see Fig. 7(c). Forasmuch as the reality gap between Jasmine robots and simulator became finally relatively low, it was also used for performing preparatory experiments with large-scale swarms, see Fig. 7(d), where the effort of performing real experiments with many attempts was especially high.

## 6 Experiments with microrobots Jasmine

Currently there are a few swarms with Jasmine robots at different institutions. In our lab, we focused primarily on exploring various ways of deriving local rules, creating artificial self-organization and different kind of emergent phenomena. There are four main approaches in developing such local rules: top-down, bottom-up, evolutionary and bio-inspired, see e.g. [15], [16], [5], [17], [18], [19] and others. The same problem can be considered in light of these four approaches: in several experiments we explored and compared their output and performance (e.g. bio-inspired [20] and tech-inspired [21] approaches). Especial attention has been



**Fig. 7.** (a) Setup in simulation; (b) The same setup in real experiments; (c) Example of a setup for exploring collective perception; (d) Example of a setup for exploring large-scale swarm.

paid to basic problems such as collective energy management, swarm communication, collective and perception and collective awareness. Several images of the performed experiments are shown in Fig. 8. Many our projects, such as [22], [23], [24], [25], [26], [27], [28] address different aspects of the performed research.

**Bottom-up approach.** The local rules are first programmed into each agent and then cyclically evaluated and redesigned, see rule-based programming [6], refining sequential program into concurrent ones [29], formal definition of coop-

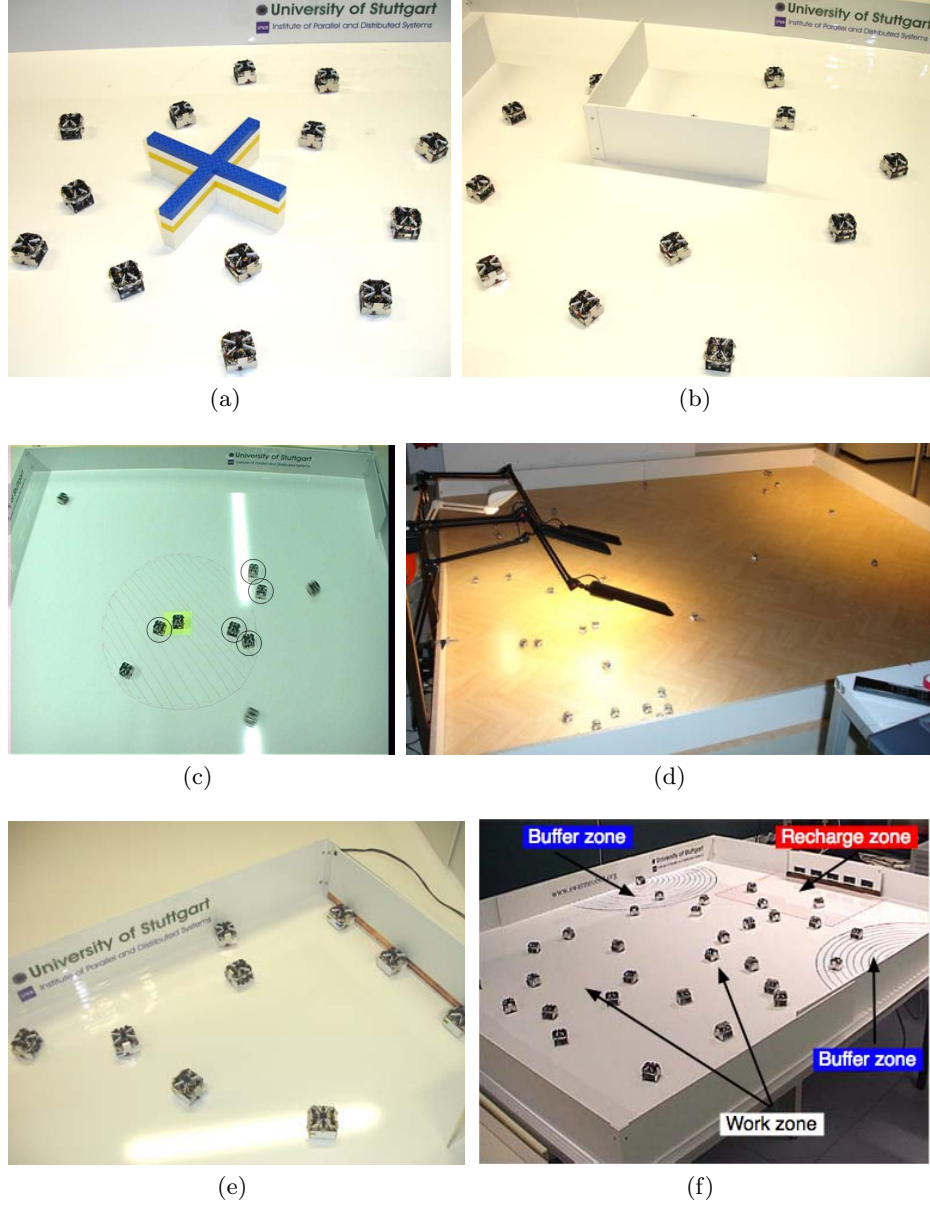
eration and coordination [30]. There are several variations of this technique: the application of optimization [31] or probabilistic [32] approaches for finding optimal rules, geometrical [33] and functional [34] considerations, the exploration of different aspects of embodiment: properties of communication [11], [35], [36], power management [37], [38] and sensing [39].

**Top-down approach.** Macroscopic behavior is formally described, see for example grammatical and semantical structures [40]. By using a formal transformation, this high-level description can be converted to low-level programs in each collective agent. The top-down approach works well in different fields of nonlinear dynamics [41], and in the application of analytical approaches for controlling collective behavior [42], [43], collective decision-making [44] and similar problems. Several optimization approaches can be used to perform top-down derivation of local rules for industrial environments [44], [45], [46], [47], [48]. In robotic systems, top-down approaches have been applied to cooperative actuation [49], [50], creation of desired behavioral patterns [14], and self-assembly processes [51], [52].

**Evolutionary approach.** The search space of a collective system contains a desired solution, that is, local rules, which are able to create a desired behavior [53]. This desired solution is described by the fitness function [54]. Applying the principles of computational evolution, the developer can find the required rules. Such an approach has been applied to foraging problems, for example [55], simple behavioral primitives [56], genetic frameworks [57], and for evolving morphology, controllers, behavior, or strategies.

**Bio-inspired approach.** Observations from social insects, animals, micro-organisms, or even humans are transferred to technical systems [58]. The number of bio-inspired works in the domain of collective systems, especially swarms, is very large. To give some examples, there are several attempts towards hormone-based regulation [59], artificial sexual reproduction [60], aggregation strategies inspired by bees [61], bio-inspired decision-making [62] and foraging [63], [20].

**Collective perception.** This problem was addressed in several works based on recognition of colors [39] and geometries [32], [64], [12]. They are related to recognition of such objects that are larger than the robot itself. For that, all robots surround an object and scan the corresponding object's surfaces (see Fig. 8(a)). The scan data provide information about surface's geometry and allow classifying the type of surface (flat, concave, convex; size of surface and so on). The classification data are exchanged between robots and matched with the distributed model of the object. However for a particular robot is important not only to recognize an object, but also to know its own position in relation to this object. This positional context cannot be obtained from the sensor data of individual robot. The idea here is that during local communication, all robots know their neighbors, and this "embodied" information can be used for estimating a position. When sequences of connected particular observations are matched with models (e.g. the model A-B-C-A-C-D and the connected particular observation A-B), these connected observations can be located in the model. In this way the robots can collectively estimate their own spatial context (see more in [32]). As



**Fig. 8.** (a) Experiments with collective perception [64]; (b) Experiments swarm communications [65], [66]; (c) Experiments with tech-inspired aggregation [67]; (d) Experiments with bio-inspired aggregation [61]; (e) Experiments with docking and recharging [21]; (f) Experiments with bio-inspired collective energy management [20].

demonstrated by experiments, even for uncomplete observations, this approach allows deriving the positions.

**Spatial information processing.** These experiments are focused on spatial information processing, such as distance/area measurement over a swarm, collective calculation of the swarm’s center of gravity [33], spatial decision making [62] and others. These works are related in many aspects to cooperative actuation [49], embodiment issues [11], [50], cooperation and functional self-organization [34]. To perform these operations, the robots have to create the dynamic communication network so that a part of them is continuously contained within the communication radius of each other. In this way they build a collective information system, which we call a ”communication street”.

**Swarm communication.** Communication approaches are integrated into other algorithms and into the software library of collective behavior. Several of these works are published in relation to IR-based [65], ZigBee [68], [57] approaches, to using context-based communication [10] and to high-level communication procedures [66], [67], [21]. Several aspects of these works are related to maintain the dynamical communication networks, optimization of energy and using the context provided by the amplitude and direction of IR signals. In this way they not only support an optimal distance for global information propagation, but also can collectively perform different spatial computations (distances, areas, centers of gravity and others).

**Feedback-based aggregation.** Aggregation was considered as a typical problem (task) for swarms and was solved via bio-inspired [61], [69], [62], [63], bottom-up [50], [39] [14], top-down [18], [9] and evolutionary [70], [56] methods. The example of aggregation was also used to compare scalability, performance and efficiency of different approaches.

**Energy-related aspects.** Energy aspects are investigated from different points of view: docking and low-level power management [37], [38], bio-inspired [20] and tech-inspired [21] behavioral strategies and energy homeostasis. We performed also several long-term experiments to explore self-sustainability of robot swarm and autonomous energy management.

**Self-assembling and multi-cellularity.** These issues are primarily related to the projects [25], [27], where we used another hardware platform [71], [72], [73], [74] or the in edited book [75]. However, initial experiments as well as swarm-related topics in these projects are still explored with Jasmine robots [76].

**Collective awareness.** These works are relatively new and related primarily to the project [28]. Here we explore mechanisms, which create awareness about a global state of the swarm on a local level without using any centralized mechanisms.

## 7 Conclusion

In this paper we presented an open-hardware project that targets a development of a cheap and reliable microrobot as well as SDK for large-scale swarms.

We demonstrated the concept of a swarm embodiment, which underlies the development of robot platform Jasmine. Its capabilities and features are briefly described and illustrated by examples. Software part and simulator are briefly described. Generally, we have shown that despite the limited hardware capabilities of microrobots, the specific construction of hardware and software parts make feasible many advanced collective properties.

The performed experiments between 2005 and 2010 are briefly listed in categories related to research topics explored in the group. However, a systematic consideration of all performed experiments and their evaluation in terms of embodiment, performance, scalability and other features still remain open and represents future works.

## 8 Acknowledgment

The *swarmrobot.org* project was started in a close collaboration with Marc Szymanski who worked at those time at the University of Karlsruhe. His enthusiasm and energy essentially contributed the success of this project.

Author and the team are supported by the followings grants: EU-IST FET project "I-SWARM", grant agreement no. 507006; EU-NMP project "Golem", grant agreement no. 033211; EU-IST FET project "SYMBRION", grant agreement no. 216342; EU-ICT project "REPLICATOR", grant agreement no. 216240; EU-ICT project "EvoBody", grant agreement no. 258334; EU-ICT FET project "Angels", grant agreement no. 231845; EU-ICT project "CoCoRo", grant agreement no. 270382.

## References

1. Kernbach, S., ed.: Handbook of Collective Robotics: Fundamentals and Challenges. Pan Stanford Publishing, Singapore (2011)
2. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence: from natural to artificial systems. Oxford University Press, New York (1999)
3. Martinoli, A.: Swarm Intelligence in autonomous collective robotics: from Tools to the analysis and synthesis of distributed control strategies PhD Thesis. Lausanne, EPFL (1999)
4. Haken, H.: Advanced synergetics. Springer-Verlag, Berlin (1983)
5. Kornienko, S., Kornienko, O., Levi, P.: Generation of desired emergent behavior in swarm of micro-robots. In Lopez de Mantaras, R., Saitta, L., eds.: Proc. of the 16th European conference on artificial intelligence (ECAI 2004), Valencia, Spain, Amsterdam: IOS Press (2004) 239–243
6. Roma, G.C., Gamble, R.F., Ball, W.E.: Formal derivation of rule-based programs. IEEE Trans. Softw. Eng. **19**(3) (1993) 277–296
7. Kornienko, S., Kornienko, O., Levi, P.: About nature of emergent behavior in micro-systems. In: Proc. of the Int. Conf. on Informatics in Control, Automation and Robotics (ICINCO 2004), Setubal, Portugal. (2004) 33–40
8. Pfeifer, R., Iida, F.: Embodied artificial intelligence: Trends and challenges. In et al. (Eds), I., ed.: Embodied artificial intelligence. Springer (2004) 1–26



9. Kornienko, S., Kornienko, O., Levi, P.: Swarm embodiment - a new way for deriving emergent behaviour in artificial swarms. In et al., P., ed.: *Autonome Mobile Systeme (AMS'05)*. (2005) 25–32
10. Kornienko, S., Kornienko, O., Levi, P.: Collective AI: context awareness via communication. In: *Proc. of the IJCAI 2005*, Edinburgh, UK. (2005) 1464–1470
11. Caselles, J.: Exploration of embodiment in real microrobotic swarm. Master Thesis, University of Stuttgart, Germany (2005)
12. Kornienko, S., Kornienko, O., Levi, P.: Minimalistic approach towards communication and perception in microrobotic swarms. In: *Proc. of the International Conference on Intelligent Robots and Systems (IROS-2005)*, Edmonton, Canada (2005) 2228–2234
13. Klein, J.: Breve Project. <http://www.spiderland.org/breve/> (2000)
14. Prieto, V.: Development of cooperative behavioural patterns for swarm robotic scenarios. Master Thesis, University of Stuttgart, Germany (2006)
15. Crespi, V., Galstyan, A., Lerman, K.: Top-down vs bottom-up methodologies in multi-agent system design. *Auton. Robots* **24**(3) (2008) 303–313
16. Isakowitz, T., Kamis, A., Koufaris, M.: Reconciling top-down and bottom-up design approaches in rmm. *SIGMIS Database* **29**(4) (1998) 58–67
17. McFarland, G.: The benefits of bottom-up design. *SIGSOFT Softw. Eng. Notes* **11**(5) (1986) 43–51
18. Kernbach, S.: *Structural Self-organization in Multi-Agents and Multi-Robotic Systems*. Logos Verlag, Berlin (2008)
19. Pizka, M., Bauer, A.: A brief top-down and bottom-up philosophy on software evolution. In: *IWPSE '04: Proceedings of the Principles of Software Evolution*, 7th International Workshop, Washington, DC, USA, IEEE Computer Society (2004) 131–136
20. Kernbach, S., Nepomnyashchikh, V., Kancheva, T., Kernbach, O.: Specialization and generalization of robotic behavior in swarm energy foraging. *Mathematical and Computer Modelling of Dynamical Systems*, DOI 10.1080/13873954.2011.601421 (2011)
21. Kernbach, S., Kernbach, O.: Collective energy homeostasis in a large-scale micro-robotic swarm. *Robotics and Autonomous Systems*, DOI 10.1016/j.robot.2011.08.001 **59** (2011) 1090–1101
22. I-Swarm: I-Swarm: Intelligent Small World Autonomous Robots for Micro-manipulation. European Union 6th Framework Programme Project No FP6-2002-IST-1 (2003-2007)
23. : GOLEM: Bio-inspired Assembly Process for Mesoscale Products and Systems. EU Project, NMP-2004-3.4.1.2-1 (2006-2009)
24. ANGELS: ANGuilliform robot with ELectric Sense, EU-project 231845, 2009-2011. European Communities (2011)
25. SYMBRION: Symbiotic Evolutionary Robot Organisms, 7th Framework Programme Project No FP7-ICT-2007.8.2. European Communities (2008-2012)
26. EVOBODY: EVOBODY: New Principles of Unbound Embodied Evolution, Project reference: 258334. European Communities (2010-2011)
27. REPLICATOR: Robotic Evolutionary Self-Programming and Self-Assembling Organisms, 7th Framework Programme Project No FP7-ICT-2007.2.1. European Communities (2008-2012)
28. CoCoRo: Collective Cognitive Robots, FP7. European Communities (2011-2013)
29. Back, R.J.R., Sere, K.: Stepwise refinement of action systems. *Structured Programming* **12** (1991) 17–30

30. Back, R.J.R., Kurki-Suonio, F.: Distributed cooperation with action systems. *ACM Trans. Program. Lang. Syst.* **10**(4) (1988) 513–554
31. Chen, X.: Optimization of communication in a swarm of micro-robots. Master Thesis, University of Stuttgart, Germany (2003)
32. Pradier, M.: Collective Classification in a Swarm of Microrobots. Master Thesis, University of Stuttgart, Germany (2005)
33. Fu, Z.: Swarm-based computation and spatial decision making. Master Thesis, University of Stuttgart, Germany (2005)
34. Warraich, O.A.: Mechanism of cooperation and functional self-organization in a swarm of micro-robots. Master Thesis, University of Stuttgart, Germany (2005)
35. Mletzko, F.: Testing and Re-Implementation of Communication Protocols for the Microrobot Jasmine. Studienarbeit, University of Stuttgart, Germany (2006)
36. Geider, R.: Development of context-based communication protocols for the micro-robot 'Jasmine'. Studienarbeit, University of Stuttgart, Germany (2006)
37. Jebens, K.: Development of a docking approach for autonomous recharging system for micro-robot "Jasmine". Student Thesis, University of Stuttgart, Germany (2006)
38. Attarzadeh, A.: Development of advanced power management for autonomous micro-robots. Master Thesis, University of Stuttgart, Germany (2006)
39. Zetterström, G.: Collaborative actuation in microrobotic swarm based on collective decision making and surfacecolor identification. Master Thesis, University of Stuttgart, Germany (2006)
40. Muscholl, M.: Interaction und Kooperation in Multiagentsystemen. PhD thesis, University of Stuttgart, Stuttgart (2001)
41. Haken, H.: *Laser theory*. Springer-Verlag, Berlin (1984)
42. Levi, P., Schanz, M., Kornienko, S., Kornienko, O.: Application of order parameter equation for the analysis and the control of nonlinear time discrete dynamical systems. *Int. J. Bifurcation and Chaos* **9**(8) (1999) 1619–1634
43. Kornienko, S., Kornienko, O.: Control of periodical motion using the synergetic concept. not published yet (1999)
44. Kornienko, O., Kornienko, S., Levi, P.: Collective decision making using natural self-organization in distributed systems. In: *Proc. of Int. Conf. on Computational Intelligence for Modelling, Control and Automation (CIMCA'2001)*, Las Vegas, USA. (2001) 460–471
45. Kornienko, S., Kornienko, O., Priese, J.: Application of multi-agent planning to the assignment problem. *Computers in Industry* **54**(3) (2004) 273–290
46. Kornienko, S., Kornienko, O., Levi, P.: Flexible manufacturing process planning based on the multi-agent technology. In: *Proc. of the 21st IASTED Int. Conf. on AI and Applications (AIA '2003)*, Innsbruck, Austria. (2003) 156–161
47. Kornienko, S., Kornienko, O., Levi, P.: Application of distributed constraint satisfaction problem to the agent-based planning in manufacturing systems. In: *Proc. of IEEE Int. Conf. on AI Systems (AIS'03)*, Divnomorsk, Russia. (2003) 124–140
48. Kornienko, S., Kornienko, O., Levi, P.: Multi-agent repairer of damaged process plans in manufacturing environment. In: *Proc. of the 8th Conf. on Intelligent Autonomous Systems (IAS-8)*, Amsterdam, NL. (2004) 485–494
49. Jiménez, M.: Cooperative actuation in a large robotic swarm. Master Thesis, University of Stuttgart, Germany (2006)
50. Mletzko, F.: Creating emergent behavior in a group of micro-robots. Master Thesis, University of Stuttgart, Germany (2006)
51. Kabir, J.: Simulation and visualization of fluid behavior in bio-inspired self-assembly processes. Master Thesis, University of Stuttgart, Germany (2008)

52. Urien, R.S.: Application and improvement of a simulation of bio-inspired self-assembly process. Master Thesis, University of Stuttgart, Germany (2009)
53. Nolfi, S., Floreano, D.: *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. The MIT Press, Cambridge, MA. / London (2000)
54. Branke, J., Schmeck, H.: Evolutionary design of emergent behavior. In Würtz, R., ed.: *Organic Computing*. Springer (2008) 123–140
55. Koza, J.: *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, Massachusetts, London, England (1992)
56. Koenig, L.: A model for developing behavioral patterns on multi-robot organisms using concepts of natural evolution. Master Thesis, University of Stuttgart, Germany (2007)
57. Nagarathinam, A.: Development of a Software Framework for Jasmine Robots using ZigBee Communication Protocol. Master Thesis, University of Stuttgart, Germany (2007)
58. Floreano, D., Husbands, P., Nolfi, S.: Evolutionary Robotics. In: *Handbook of Robotics*. Springer Verlag, Berlin (2008)
59. Speidel, G.: Artificial cell differentiation in a multi-robot organisms through gene regulation. Master Thesis, University of Stuttgart, Germany (2008)
60. Schwarzer, C.: Investigation of Evolutionary Reproduction in a Robot Swarm. Master Thesis, University of Stuttgart, Germany (2008)
61. Kernbach, S., Thenius, R., Kernbach, O., Schmickl, T.: Re-embodiment of honeybee aggregation behavior in artificial micro-robotic system. *Adaptive Behavior* **17**(3) (2009) 237–259
62. Häbe, D.: Bio-inspired approach towards collective decision making in robotic swarms. Master Thesis, University of Stuttgart, Germany (2007)
63. Kancheva, T.: Adaptive role dynamics in energy foraging behavior of a real micro-robotic swarm. Master Thesis, University of Stuttgart, Germany (2007)
64. Kornienko, S., Kornienko, O., Constantinescu, C., Pradier, M., Levi, P.: Cognitive micro-agents: individual and collective perception in microrobotic swarm. In: *Proc. of the IJCAI-05 Workshop on Agents in real-time and dynamic environments*, Edinburgh, UK. (2005) 33–42
65. Kornienko, S., Kornienko, O.: IR-based communication and perception in micro-robotic swarms. In: *IROS 2005*, Edmonton, Canada. (2005)
66. Kernbach, S.: Three Cases of Connectivity and Global Information Transfer in Robot Swarms. *ArXiv e-prints* (2011arXiv1109.4221K) (2011)
67. Kernbach, S.: Diffusion of information in robot swarms. *ArXiv e-prints* (arXiv:1110.5183v1) (2011)
68. Beurer, D.: Design and tests of ZigBee communication system for robot organisms. Master Thesis, University of Stuttgart, Germany (2008)
69. Schmickl, T., Thenius, R., Moeslinger, C., Radspieler, G., Kernbach, S., Szymanski, M., Crailsheim, K.: Get in touch: cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems* **18** (2009) 133–155 10.1007/s10458-008-9058-5.
70. Koenig, L., Jebens, K., Kernbach, S., Levi, P.: Stability of on-line and on-board evolving of adaptive collective behavior. In: *Proc. of the EUROS 2008*, Prague, Czech Republic. (2007)
71. Kernbach, S., Meister, E., Scholz, O., Humza, R., Liedke, J., Ricotti, L., Jemai, J., Havlik, J., Liu, W.: Evolutionary robotics: The next-generation-platform for on-line and on-board artificial evolution. In: *IEEE Congress on Evolutionary Computation, CEC '09*. (2009) 1079 –1086

72. Kernbach, S., Ricotti, L., Liedke, J., Corradi, P., Rothermel, M.: Study of macroscopic morphological features of symbiotic robotic organisms. In: Proceedings of the workshop on self-reconfigurable robots, IROS08, Nice. (2008) 18–25
73. Kernbach, S., Meister, E., Schlachter, F., Jebens, K., Szymanski, M., Liedke, J., Laneri, D., Winkler, L., Schmickl, T., Thenius, R., Corradi, P., Ricotti, L.: Symbiotic robot organisms: REPLICATOR and SYMBRION projects. In: Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems. PerMIS '08, New York, NY, USA, ACM (2008) 62–69
74. Kernbach, S., Hamann, H., Stradner, J., Thenius, R., Schmickl, T., Crailsheim, K., Rossum, A.C.v., Sebag, M., Bredeche, N., Yao, Y., Baele, G., Peer, Y.V.d., Timmis, J., Mohktar, M., Tyrrell, A., Eiben, A.E., McKibbin, S.P., Liu, W., Winfield, A.F.T.: On adaptive self-organization in artificial robot organisms. In: Proceedings of the 2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns. COMPUTATIONWORLD '09, Washington, DC, USA, IEEE Computer Society (2009) 33–43
75. Levi, P., Kernbach, S., eds.: Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution. Springer Verlag (2010)
76. Kornienko, S., Kornienko, O., Nagarathinam, A., Levi, P.: From real robot swarm to evolutionary multi-robot organism. In: Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. (2007) 1483–1490